Listing of the Claims:

The following is a complete listing of all the claims in the application, with an indication of the status of each:

1. (Canceled)

2. (Currently amended A bridge apparatus ~~according to claim 1, further~~ comprising:

a first device driver unit for controlling a first interface unit connected to a first network;

a second device driver unit for controlling a second interface unit connected to a second network;

a bridging unit for performing a bridging process;

a middleware unit, inserted between the bridging unit and the first device driver unit, that includes

a transmitter for performing priority processing for a relay of a frame from the bridging unit to the first device driver unit;

a cache table in which session data having high priorities are preregistered; and

a plurality of first FIFO queues corresponding to priorities,

wherein the transmitter includes

a header comparator for, upon the reception of a transmission request for the frame to be relayed from the bridging unit to the first device driver unit, searching the cache table and extracting a priority based on headers included in a second to a fourth OSI layer of the frame, and for adding the transmission request to one of the first FIFO queues in accordance with the priority that is extracted, and

a synthesization unit for, in accordance with a priority for the first FIFO queue, outputting the transmission request from the first FIFO queue to the first device driver unit.

3. (Currently amended) A bridge apparatus ~~according to claim 1, further~~ comprising:

a first device driver unit for controlling a first interface unit connected to a first network;

a second device driver unit for controlling a second interface unit connected to a second network;

a bridging unit for performing a bridging process;

a middleware unit, inserted between the bridging unit and the first device driver unit, that includes

a transmitter for performing priority processing for a relay of a frame from the bridging unit to the first device driver unit;

a first cache table, in which first session data are predesignated;

a second cache table, used when a session is established;

a first FIFO queue; and

a second FIFO queue,

wherein the middleware unit includes

a first header comparator for, when a transmission request is issued for the frame to be relayed from the bridging unit to the first device, extracting second session data from headers of a second to a fourth OSI layer in the frame and, when the second session data are registered in the second cache table, adding the transmission request to the first FIFO queue; for, when the second session data are registered neither in the first cache table nor in the second cache table and the frame to be relayed is a specific, predesignated frame, registering the second session data in the second cache table and adding the transmission request to the first FIFO queue; for, when the second session data are registered in the first cache table but not in the second cache table and the frame is not a specific, predesignated frame, adding the transmission request to the second FIFO buffer; or for, when the second session data are registered neither in the first nor the second cache tables, adding the transmission request to the second FIFO queue, and

a synthesization unit for outputting to the first device driver unit, in the named order, the transmission requests in the first FIFO queue and in the second FIFO queue.

4. (Original) A bridge apparatus according to claim 3, wherein the middleware

unit includes:

the header comparator for activating a frame monitor timer when the second session data are not registered in the second cache table but are registered in the first cache table, and when the frame to be relayed is a specific predesignated frame; and

a monitor unit for, when the value of the frame monitor timer has reached a predetermined value, deleting from the second cache table the second session data that correspond to the frame monitor timer.

5. (Canceled)

6. (Currently amended) A bridge apparatus ~~according to claim 5, further~~ comprising:

first device driver unit for controlling a first interface unit connected to a first network;

a second device driver unit for controlling a second interface unit connected to a second network;

a bridging unit for determining whether the address of a frame received from the first or the second network is registered in an address table, and for performing bridging processing for the frame;

a middleware unit including

a transmitter, inserted between the bridging unit and the first device driver unit, for performing the priority processing for a first frame to be relayed from the bridging unit to the first device driver unit, and

a receiver, inserted between the bridging unit and the second device driver unit, for performing the priority processing for a second frame to be relayed from the first device driver unit to the bridging unit;

a cache table in which session data are preregistered;

a first FIFO queue for the transmitter; and

a second FIFO queue for the receiver,

wherein the transmitter includes

a first header comparator for, upon the reception of a transmission request for the first frame to be relayed from the bridging unit to the first device

driver unit, searching the cache table and extracting a first priority, based on headers that are included in a second to a fourth OSI layer in the first frame, and for, in accordance with the first priority, adding the transmission request for the first frame to the first FIFO queue, and

a first synthesization unit for transmitting, in accordance with the first priority, the transmission request from the first FIFO queue to the first device driver unit,

and wherein the receiver includes

a second header comparator for, upon the reception of a bridging request for the second frame, searching the cache table and extracting a second priority, based on headers that are included in a second to a fourth OSI layer in the second frame, and, for, in accordance with the second priority, adding the bridging request for the second frame to the second FIFO queue, and

a second synthesization unit for transmitting, in accordance with the second priority, the bridging request from the second FIFO queue to the bridging unit.

7. (Original) A bridge apparatus comprising:

a first device driver unit for controlling a first interface unit connected to a first network;

a second device driver unit for controlling a second interface unit connected to a second network;

a bridging unit for examining the address of a frame received from the first or the second network to determine whether the address is registered in an address table, and for performing bridging processing for the frame;

a middleware unit, inserted between the bridging unit and the first device driver unit;

a first cache table, in which first session data having a high priority are predesignated;

a second cache table, used when a session is established;

a first FIFO queue;

a second FIFO queue;

a third FIFO queue; and

a fourth FIFO queue,

wherein the middleware unit includes

a first header comparator for, when a transmission request is issued for a first frame to relayed from the bridging unit to the first device driver unit, extracting second session data from headers of a second to a fourth OSI layer in the first frame and, when the second session data are registered in the second cache table, adding the transmission request to the first FIFO queue; for, when the second session data are registered in the first cache table but not in the second cache table and the first frame to be relayed is a specific, predesignated frame, registering the second session data in the second cache table and adding the transmission request to the first FIFO queue; for, when the second session data are registered in the first cache table but not in the second cache table and the first frame is not a specific, predesignated frame, adding the transmission request to the second FIFO buffer; or for, when the second session data are registered neither in the first nor the second cache tables, adding the transmission request to the second FIFO queue,

a first synthesization unit for outputting to the first device driver unit, in the named order, the transmission requests in the first FIFO queue and in the second FIFO queue,

a second header comparator for, when a bridging request is issued for a second frame to be relayed from the first device driver unit to the bridging unit, extracting third session data from headers of a second to a fourth OSI layer in the second frame and, when the third session data are registered in the second cache table, adding the bridging request to the third FIFO queue; for, when the third session data are registered in the first cache table but not in the second cache table and the second frame to be relayed is a specific, predesignated frame, registering the third session data in the second cache table and adding the bridging request to the third FIFO queue; for, when the third session data are registered in the first cache table but not in the second cache table and the second frame is not a specific, predesignated frame, adding the bridging request to the fourth FIFO queue; or for, when the second session data are registered neither in the first nor in the second cache tables, adding the bridging request to the fourth FIFO queue, and

a second synthesization unit for outputting to the bridging unit, in

the named order, the bridging requests in the third FIFO queue and in the fourth FIFO queue.

8. (Original) A bridge apparatus according to claim 7, wherein the middleware unit includes:

the first header comparator for activating a first frame monitor timer when the second session data are not registered in the second cache table but are registered in the first cache table, and when the first frame to be relayed is a specific predesignated frame;

the second header comparator for activating a second frame monitor timer when the third session data are not registered in the second cache table but are registered in the first cache table, and when the second frame to be relayed is a specific predesignated frame; and

a monitor unit for, when the value of the first frame monitor timer has reached a predetermined value, deleting from the second cache table the second session data that correspond to the first frame monitor timer, and for, when the value of the second frame monitor timer has reached a predetermined value, deleting from the second cache table the third session data that correspond to the second frame monitor timer.

9. (Original) A bridge apparatus comprising:

a first device driver unit for controlling a first interface unit connected to a first network;

a second device driver unit for controlling a second interface unit connected to a second network;

a bridging unit for determining whether the address of a frame received from the first or the second network is registered in an address table, and for performing bridging processing for the frame;

a first middleware unit including

a first transmitter, inserted between the bridging unit and the first device driver unit, for performing the priority processing for a first frame to be relayed from the bridging unit to the first device driver unit, and

a first receiver, inserted between the bridging unit and the first

device driver unit, for performing the priority processing for a second frame to be relayed from the first device driver unit to the bridging unit; and

a second middleware unit including

a second transmitter, inserted between the bridging unit and the second device driver unit, for performing the priority processing for a third frame to be relayed from the bridging unit to the second device driver unit, and

a second receiver, inserted between the bridging unit and the second device driver unit, for performing the priority processing for a third frame to be relayed from the second device driver unit to the bridging unit.

10. (Original) A bridge apparatus according to claim 9, further comprising:

a cache table in which session data having a high priority are preregistered;

a transmission FIFO queue, used for the second transmitter and corresponding to a priority; and

a reception FIFO queue, used for the second receiver and corresponding to a priority,

wherein the second transmitter includes

a transmission request header comparator for, upon the reception of a transmission request for the third frame to be relayed from the bridging unit to the second device driver unit, searching the cache table and, based on headers that are included in a second to a fourth OSI layer in the third frame, extracting a transmission priority and, in accordance with the transmission priority, adding the transmission request for the third frame to the transmission FIFO queue, and

a transmission request synthesization unit for transmitting, in accordance with the transmission priority, the transmission request from the transmission FIFO queue to the second device driver unit,

and wherein the second receiver includes

a bridging request header comparator for, upon the reception of a bridging request for the fourth frame to be relayed from the second device driver unit to the bridging unit, searching the cache table and, based on headers that are included in a second to a fourth OSI layer in the fourth frame, extracting a reception priority and, in accordance with the reception priority, adding the bridging request for the fourth frame to the reception FIFO queue, and

a bridging request synthesization unit for, in accordance with the reception priority, transmitting the bridging request from the reception FIFO queue to the bridging unit.

11. (Original) A bridge apparatus comprising:

a first device driver unit for controlling a first interface unit connected to a first network;

a second device driver unit for controlling a second interface unit connected to a second network;

a bridging unit for performing bridging processing;

a middleware unit, inserted between the bridging unit and the first and second device driver units;

a first cache table, in which first session data are predesignated;

a second cache table, used when a session is established;

a third cache table, in which fourth session data are predesignated;

a fourth cache table, used when a session is established;

a first FIFO queue;

a second FIFO queue;

a third FIFO queue;

a fourth FIFO queue;

a fifth FIFO queue;

a sixth FIFO queue;

a seventh FIFO queue; and

an eighth FIFO queue,

wherein the middleware unit includes

a first header comparator for, when a transmission request is issued for a first frame to relayed from the bridging unit to the first device driver unit, extracting second session data from headers of a second to a fourth OSI layer in the first frame and, when the second session data are registered in the second cache table, adding the transmission request for the first frame to the first FIFO queue; for, when the second session data are registered in the first cache table but not in the second cache table and the first frame to be relayed is a specific, predesignated frame, registering the second session data in the second cache table

and adding the transmission request to the first FIFO queue; for, when the second session data are registered in the first cache table but not in the second cache table and the first frame is not a specific, predesignated frame, adding the transmission request to the second FIFO buffer; or for, when the second session data are registered neither in the first nor the second cache tables, adding the transmission request to the second FIFO queue,

a first synthesization unit for outputting to the first device driver unit, in the named order, the transmission requests for the first frame in the first FIFO queue and in the second FIFO queue,

a second header comparator for, when a bridging request is issued for a second frame to be relayed from the first device driver unit to the bridging unit, extracting third session data from headers of a second to a fourth OSI layer in the second frame and, when the third session data are registered in the second cache table, adding the bridging request for the second frame to the third FIFO queue; for, when the third session data are registered in the first cache table but not in the second cache table and the second frame to be relayed is a specific, predesignated frame, registering the third session data in the second cache table and adding the bridging request to the third FIFO queue; for, when the third session data are registered in the first cache table but not in the second cache table and the second frame is not a specific, predesignated frame, adding the bridging request to the fourth FIFO queue; or for, when the second session data are registered neither in the first nor in the second cache tables, adding the bridging request to the fourth FIFO queue,

a second synthesization unit for outputting to the bridging unit, in the named order, the bridging requests for the second frame in the third FIFO queue and in the fourth FIFO queue,

a third header comparator for, when a transmission request is issued for a third frame to relayed from the bridging unit to the second device driver unit, extracting fifth session data from headers of a second to a fourth OSI layer in the third frame and, when the fifth session data are registered in the fourth cache table, adding the transmission request for the third frame to the fifth FIFO queue; for, when the fifth session data are registered in the third cache table but not in the fourth cache table and the third frame to be relayed is a specific,

predesignated frame, registering the fifth session data in the fourth cache table and adding the transmission request to the fifth FIFO queue; for, when the fifth session data are registered in the third cache table but not in the fourth cache table and the third frame is not a specific, predesignated frame, adding the transmission request to the sixth FIFO buffer; or for, when the fifth session data are registered neither in the third nor the fourth cache tables, adding the transmission request to the sixth FIFO queue,

a third synthesization unit for outputting to the second device driver unit, in the named order, the transmission requests for the third frame in the fifth FIFO queue and in the sixth FIFO queue,

a fourth header comparator for, when a bridging request is issued for a fourth frame to be relayed from the second device driver unit to the bridging unit, extracting sixth session data from headers of a second to a fourth OSI layer in the fourth frame and, when the sixth session data are registered in the fourth cache table, adding the bridging request for the fourth frame to the seventh FIFO queue; for, when the sixth session data are registered in the third cache table but not in the fourth cache table and the fourth frame to be relayed is a specific, predesignated frame, registering the sixth session data in the fourth cache table and adding the bridging request to the seventh FIFO queue; for, when the sixth session data are registered in the third cache table but not in the fourth cache table and the fourth frame is not a specific, predesignated frame, adding the bridging request to the eighth FIFO queue; or for, when the sixth session data are registered neither in the third nor in the fourth cache tables, adding the bridging request to the eighth FIFO queue, and

a fourth synthesization unit for outputting to the bridging unit, in the named order, the bridging requests for the fourth frame in the seventh FIFO queue and in the eighth FIFO queue.

12. (Original) A bridge apparatus according to claim 11, further comprising:

a first monitor timer;

a second monitor timer;

a third monitor timer; and

a fourth monitor timer,

wherein the middleware unit includes

a monitor unit for deleting the second session data from the second cache table when a specific value is reached in the first monitor timer, for deleting the third session data from the second cache table when a specific value is reached in the second monitor timer, for deleting the fourth session data from the fourth cache table when a specific value is reached in the third monitor timer, and for deleting the fifth session data from the fourth cache table when a specific value is reached in a fourth monitor timer.

13. (Original) A bridge apparatus comprising:

a first device driver unit for controlling a first interface unit connected to a first network;

a second device driver unit for controlling a second interface unit connected to a second network;

a bridging unit for performing bridging processing;

a cache table in which session data having a high priority are stored;

a first FIFO queue corresponding to a priority;

a second FIFO queue corresponding to a priority,

wherein the bridging unit includes

a bridging processor connected to the first device driver unit and the second device driver unit,

a first header comparator for, when a first bridging request for a first frame to be relayed is received from the first device driver unit, searching the cache table and extracting a first priority for the first bridging request, based on headers that are included in a second to a fourth OSI layer in the first frame, and for, in accordance with the first priority, adding the first bridging request to the first FIFO queue,

a first synthesization unit for transmitting, in accordance with the first priority, the first bridging request from the first FIFO queue to the bridging processor,

a second header comparator for, when a second bridging request for a second frame to be relayed is received from the second device driver unit, searching the cache table and extracting a second priority for the second bridging

request, based on headers that are included in a second to a fourth OSI layer in the second frame, and for, in accordance with the second priority, adding the second bridging request to the second FIFO queue, and

a second synthesization unit for transmitting, in accordance with the second priority, the second bridging request from the second FIFO queue to the bridging processor.

14. (Original) A bridge apparatus comprising:

a first device driver unit for controlling a first interface unit connected to a first network;

a second device driver unit for controlling a second interface unit connected to a second network;

a bridging processor for performing bridging processing;

a first cache table, in which first session data having a high priority are predesignated;

a second cache table, used when a session is established;

a first FIFO queue;

a second FIFO queue;

a third FIFO queue; and

a fourth FIFO queue,

wherein the middleware unit includes

a first header comparator for, when a bridging request is issued for a first frame to relayed from the first device driver unit to the bridging processor, extracting second session data from headers of a second to a fourth OSI layer in the first frame and, when the second session data are registered in the second cache table, adding the bridging request for the first frame to the first FIFO queue; for, when the second session data are registered in the first cache table but not in the second cache table and the first frame to be relayed is a specific, predesignated frame, registering the second session data in the second cache table and adding the bridging request to the first FIFO queue; for, when the second session data are registered in the first cache table but not in the second cache table and the first frame is not a specific, predesignated frame, adding the bridging request to the second FIFO buffer; or for, when the second session data are registered neither in

the first nor the second cache tables, adding the bridging request to the second FIFO queue,

a first synthesization unit for outputting to the bridging processor, in the named order, the bridging requests for the first frame in the first FIFO queue and in the second FIFO queue,

a second header comparator for, when a bridging request is issued for a second frame to be relayed from the second device driver unit to the bridging processor, extracting third session data from headers of a second to a fourth OSI layer in the second frame and, when the third session data are registered in the second cache table, adding the bridging request for the second frame to the third FIFO queue; for, when the third session data are registered in the first cache table but not in the second cache table and the second frame to be relayed is a specific, predesignated frame, registering the third session data in the second cache table and adding the bridging request to the third FIFO queue; for, when the third session data are registered in the first cache table but not in the second cache table and the second frame is not a specific, predesignated frame, adding the bridging request to the fourth FIFO queue; or for, when the second session data are registered neither in the first nor in the second cache tables, adding the bridging request to the fourth FIFO queue, and

a second synthesization unit for outputting to the bridging processor, in the named order, the bridging requests for the second frame in the third FIFO queue and in the fourth FIFO queue.

15. (Original) A bridge apparatus according to claim 14, further comprising:

a first monitor timer;

a second monitor timer; and

a monitor unit for deleting the second session data from the second cache table when a specific value is reached in the first monitor timer, for deleting the third session data from the second cache table when a specific value is reached in the second monitor timer, for deleting the fourth session data from the fourth cache table when a specific value is reached in the third monitor timer, and for deleting the fifth session data from the fourth cache table when a specific value is reached in a fourth monitor timer.

16. (Original) A bridge apparatus according to claim 3, wherein the specific frame is a frame, including an RTP frame, defined by a communication protocol equal to or higher than a fifth OSI layer.

17. (Original) A bridge apparatus according to claim 2, wherein the session data include an MAC address that is pertinent to the second OSI layer of the frame, a protocol number and an IP address that are pertinent to the third OSI layer, and a port number that is pertinent to the fourth OSI layer.

18. (Original) A bridge apparatus according to claim 3, wherein the first session data include an MAC address that is pertinent to the second OSI layer of the frame, a protocol number and an IP address that are pertinent to the third OSI layer, and a port number that is pertinent to the fourth OSI layer.

19. (Original) A bridge apparatus according to claim 7, wherein the second and third session data include an MAC address that is pertinent to the second OSI layer of the frame, a protocol number and an IP address that are pertinent to the third OSI layer, and a port number that is pertinent to the fourth OSI layer.

20. (Original) A bridge apparatus according to claim 11, wherein the fourth session data include an MAC address that is pertinent to the second OSI layer of the frame, a protocol number and an IP address that are pertinent to the third OSI layer, and a port number that is pertinent to the fourth OSI layer.

21. (Original) A bridge apparatus according to claim 11, wherein the fifth and sixth session data include an MAC address that is pertinent to the second OSI layer of the frame, a protocol number and an IP address that are pertinent to the third OSI layer, and a port number that is pertinent to the fourth OSI layer.

22. (Canceled)

23. (Canceled)

24. (Original) A bridge method, for a bridge apparatus that relays a second network and a first network, comprising steps of:

receiving a specific frame to be relayed from the second network to the first network and extracting session data from headers of a second to a fourth OSI layer in the specific frame;

when the session data satisfy a predetermined condition, increasing a bridging priority for the specific frame and performing bridging processing;

thereafter, when the session data extracted from the specific frame satisfy a predetermined condition, increasing a transmission priority for the specific frame and transmitting the specific frame to the first network;

receiving a specific frame to be relayed from the first network to the second network, and extracting session data from headers of a second to a fourth OSI layer in the specific frame;

when the session data satisfy a predetermined condition, increasing a bridging priority for the specific frame and performing bridging processing;

thereafter, extracting session data from the headers of the second to the fourth OSI layer in the specific frame; and

when the session data extracted from the specific frame satisfy a predetermined condition, increasing a transmission priority for the specific frame and transmitting the specific frame to the second network.

25. (Original) A bridge method for a bridge apparatus that relays frames for a second network and a first network comprising the steps of:

receiving from the second network a specific frame to be relayed to the first network;

when session data extracted from headers of a second to a fourth OSI layer in the specific frame satisfy a predetermined condition, providing a higher priority for the specific frame in a bridging queue, performing bridging processing and transmitting the specific frame to the first network;

receiving from the first network a specific frame addressing a transmission destination connected to the second network;

when session data extracted from headers of a second to a fourth OSI layer in the specific frame satisfy a predetermined condition, providing a higher priority

for the specific frame in a bridging queue, performing bridging processing, and transmitting the specific frame to the second network.

26. (Original) A bridge method, for a bridge apparatus that comprises a device driver for controlling interface units connected to a plurality of networks, a bridging unit, for comparing the address of a frame received via each of the networks with each MAC address registered in an address table and for performing bridging processing for the frame, and a middleware unit, for controlling the interface unit between the bridging unit and the device driver, comprising steps of:

when the bridging unit issues transmission requests for relaying specific sequential frames to a predetermined transmission destination, upon the reception of a first transmission request, the middleware unit, for confirming header data included in a first specific frame, extracting session data from headers of a second to a fourth OSI layer of the first specific frame, registering the session data in a cache table, increasing a transmission priority for the first specific frame, and issuing a transmission request to the device driver; and

when the succeeding specific frames are to be sequentially transmitted, the middleware unit comparing the session data registered in the cache table with session data extracted from the second to the fourth OSI layers of the succeeding specific frames, transmission priorities being increased for specific succeeding frames, and transmission requests being output to the device driver.

27. (Original) A bridge method, according to claim 26, further comprising steps of:

when the device driver issues bridging requests for specific sequential frames, which are output by a predetermined transmission source and which are defined by a communication protocol as being equal to or higher than a fifth OSI layer, upon the reception of a first bridging request, the middleware unit confirming header data in a first specific frame, extracting session data from headers for a second to a fourth OSI layer in the first specific frame, registering the session data in a cache table, increasing a bridging priority for the first specific frame and issuing a bridging request to the bridging unit; and,

when succeeding specific frames are to be sequentially received, the middleware unit comparing the session data registered in the cache table with session data extracted from second to fourth OSI layers in the succeeding specific frames, increasing bridging priorities for the succeeding specific frames, and outputting bridging requests to the bridging unit.

28. (Original) A bridge method, according to claim 26, wherefor, when session data are registered in the cache table, monitoring of the session data is continuously performed by the middleware unit until a predetermined period of time has elapsed and no transmission request has been received from the bridging unit for a frame having the session data.

29. (Original) A bridge method, according to claim 27, wherefor the middleware unit includes a monitor timer; and wherefor, when a value held by the monitor timer reaches a predetermined value, session data are deleted from the cache table by the middleware unit.

30. (Currently amended) A bridge method for a bridge apparatus that relays frames for a second network and a first network comprising the steps of:

receiving from the second network a specific frame to be relayed to the first network;

when session data extracted session data from headers of a second to a fourth OSI layer of the specific frame satisfy a predetermined condition, providing a higher priority for the specific frame in a transmission queue and transmitting the specific frame to the first network; and

~~, according to claim 22, further comprising a step of:~~

receiving the specific frame that is to be relayed by the second network to the first network and that is defined by a communication protocol equal to or higher than the fifth OSI layer and that includes an RTP frame.

31. (Currently amended) A bridge method for a bridge apparatus that relays frames for a second network and a first network comprising the steps of:

receiving from the second network a specific frame to be relayed to the

first network;

when session data extracted session data from headers of a second to a fourth OSI layer of the specific frame satisfy a predetermined condition, providing a higher priority for the specific frame in a transmission queue and transmitting the specific frame to the first network; and

~~, according to claim 22, further comprising a step of:~~

extracting the session data, which include an MAC address pertinent to the second OSI layer for the specific frame, a protocol number and an IP address pertinent to the third OSI layer, and a port number pertinent to the fourth OSI layer.